Topics for Midterm

- Copy Constructors
- Inheritance
- Single Inheritance
- Abstract Base Class
- Pure Virtual Functions
- Dynamic Binding
- Oops!
- goof"
class datn: public node
data

3: 
- copy tree into LLL
- remove leaf / leaf / small leaf / copy tree
- BST

- remove at either end
- make a copy

1) LLL - Linear
- make a copy
- remove the last 2 items
- removing at end
- adding at end

LLL - Singly Linked List
Client

char str[10];
char dummy;
Data member

strcpy(str, dummy);

return str[index];

// Add Boundary Check

for (ind = 0; str[ind] != char f String [greater C (initial)]

str[c] = 'a';

}
(\mathcal{L} + (\mathcal{L} \mathcal{L} + \mathcal{L}) \mathcal{L} \mathcal{L})^* \\
\text{multi-dimensional CI}
underly & operator() (int row, int col)
    ↑
underlying data &
    float &

\[
\text{matrix}(i, j) \times \text{float}
\]
Increment

Operator or

Must be an object
of class

Value

rvalue

++
i

i++

Member

Class

count & operator++ ( _______ ); // prefix

count & operator++ ( int _______ ); // postfix
```cpp
counter counter::operator++(int)
{
    counter temp(*this);
    data += 1;
    return temp;
}
```
This $\Rightarrow$ checking: $\text{operator} = (\text{cast student & s1})$.
Student obj = Student obj2
Client

Overloaded
Student
explicity
\[
\begin{align*}
\text{not} & \quad \text{implied}
\end{align*}
\]
f (char *)
f (char [*])
Node

Strut node

Char * dup

unsigned

You can only put I name.h the non
function

! = int (f)?

! = int (f)?

! = cast "C\47" char

Explicit Type Conversion
String

Name

Conversion

Make

String

Operator (c)

Property

String

Number

Operation

Get

If positive

Get

(name ("hi", "hi")

0) = name

string top

top = &str

Return by value

MUST